	L #	Hits	Search Text	DBs
1	L1	1130	(fenc$3 synch$ barrier) near10 load near10 (instruction operation)	USPAT; US-PGPUB
2	L2	21	1 near50 (load near2 buffer)	USPAT; US-PGPUB
3	L3	2	1 near50 (load near10 global$2)	USPAT; US-PGPUB
4	L4	344	(fenc$3 synch$ barrier) near10 load near10 (instruction operation)	EPO; JPO; DERWENT; IBM_TDB
5	L5	1	4 near50 (load near2 buffer)	EPO; JPO; DERWENT; IBM_TDB
6	L6	1	4 near50 (load near10 global$2)	EPO; JPO; DERWENT; IBM_TDB
7	L7	4	3 5 6	USPAT; US-PGPUB ; EPO; JPO; DERWENT; IBM_TDB
8	L8	1197	(fenc$3 sync$ barrier) near10 load near10 (instruction operation)	USPAT; US-PGPUB
9	L9	21	8 near50 (load near2 buffer)	USPAT; US-PGPUB
10	L12	2	4 and (load near2 buffer)	EPO; JPO; DERWENT; IBM_TDB
11	L14	52	8 and (load near2 buffer) not 2	USPAT; US-PGPUB
12	L16	38	8 and (load near10 global$2) not (9 14)	USPAT; US-PGPUB
13	L15	46	8 and (load near10 global$2)	USPAT; US-PGPUB
14	L19	1	4 and (load near10 global$2)	EPO; JPO; DERWENT; IBM_TDB
15	L18	2	4 and (load near2 buffer)	EPO; JPO; DERWENT; IBM_TDB
16	L21	1	4 and (load$3 near10 global$2)	EPO; JPO; DERWENT; IBM_TDB
17	L22	4	8 and (load$3 near10 global$2) not 15	USPAT; US-PGPUB

CONTENTS OF A PORTION OF PROCESSING ELEMENT 16	SUB-STEPS 1, 2, 3	SUB-STEPS 4, 5, 6	SUB-STEPS 7, 8, 9	FINAL VECTOR AGGREGATION RESULT
CONTENTS OF PROCESSING ELEMENT #0	A			
CONTENTS OF PROCESSING ELEMENT #1	B ⊕	A+B		
CONTENTS OF PROCESSING ELEMENT #2	C			
CONTENTS OF PROCESSING ELEMENT #3	D ⊕	C+D ⊕	A+B+C+D	
CONTENTS OF PROCESSING ELEMENT #4	E			
CONTENTS OF PROCESSING ELEMENT #5	F ⊕	E+F		
CONTENTS OF PROCESSING ELEMENT #6	G			
CONTENTS OF PROCESSING ELEMENT #7	H ⊕	G+H ⊕	E+F+G+H ⊕	A+B+C+D+E+F+G+H

PRIOR ART APPROACH USING SINGLE INSTRUCTION MULTIPLE DATA (SIMD) PROCESSOR INSTRUCTION STEPS:

1. COPY VECTOR
2. SHIFT COPY 1 PE DOWNWARD
3. ADD SHIFTED COPY TO VECTOR
4. COPY TO VECTOR
5. SHIFT COPY 2 PEs DOWNWARD
6. ADD SHIFTED COPY TO VECTOR
7. COPY VECTOR
8. SHIFT COPY 4 PEs DOWNWARD
9. ADD SHIFTED COPY TO VECTOR

FIG.5-12
-PRIOR ART-

	Document ID	U	Title	Current OR
1	US 20030084259 A1	☐	MFENCE and LFENCE micro-architectural implementation method and system	711/163
2	US 20020199067 A1	☒	System and method for high performance execution of locked memory instructions in a system with distributed memory and a restrictive memory model	711/145
3	US 20020112146 A1	☒	Method and apparatus for synchronizing load operation	712/219
4	US 20020087810 A1	☒	System and method for high performance execution of locked memory instructions in a system with distributed memory and a restrictive memory model	711/145
5	US 20010042187 A1	☒	VARIABLE ISSUE-WIDTH VLIW PROCESSOR	712/2
6	US 6615338 B1	☒	Clustered architecture in a VLIW processor	712/24
7	US 6611900 B2	☒	System and method for high performance execution of locked memory instructions in a system with distributed memory and a restrictive memory model	711/145
8	US 6463511 B2	☒	System and method for high performance execution of locked memory instructions in a system with distributed memory and a restrictive memory model	711/145
9	US 6397354 B1	☒	Method and apparatus for providing external access to signals that are internal to an integrated circuit chip package	714/34
10	US 6374370 B1	☒	Method and system for flexible control of BIST registers based upon on-chip events	714/39
11	US 6279100 B1	☒	Local stall control method and structure in a microprocessor	712/24
12	US 6009539 A	☒	Cross-triggering CPUs for enhanced test operations in a multi-CPU computer system	714/30
13	US 6003107 A	☒	Circuitry for providing external access to signals that are internal to an integrated circuit chip package	710/316
14	US 5966515 A	☒	Parallel emulation system and method	703/21
15	US 5956477 A	☒	Method for processing information in a microprocessor to facilitate debug and performance monitoring	714/30
16	US 5956476 A	☒	Circuitry and method for detecting signal patterns on a bus using dynamically changing expected patterns	714/30
17	US 5887003 A	☒	Apparatus and method for comparing a group of binary fields with an expected pattern to generate match results	714/736
18	US 5881224 A	☒	Apparatus and method for tracking events in a microprocessor that can retire more than one instruction during a clock cycle	714/47
19	US 5881017 A	☒	Input comparison circuitry and method for a programmable state machine	714/30
20	US 5880671 A	☒	Flexible circuitry and method for detecting signal patterns on a bus	340/146.2
21	US 5867644 A	☐	System and method for on-chip debug support and performance monitoring in a microprocessor	714/39

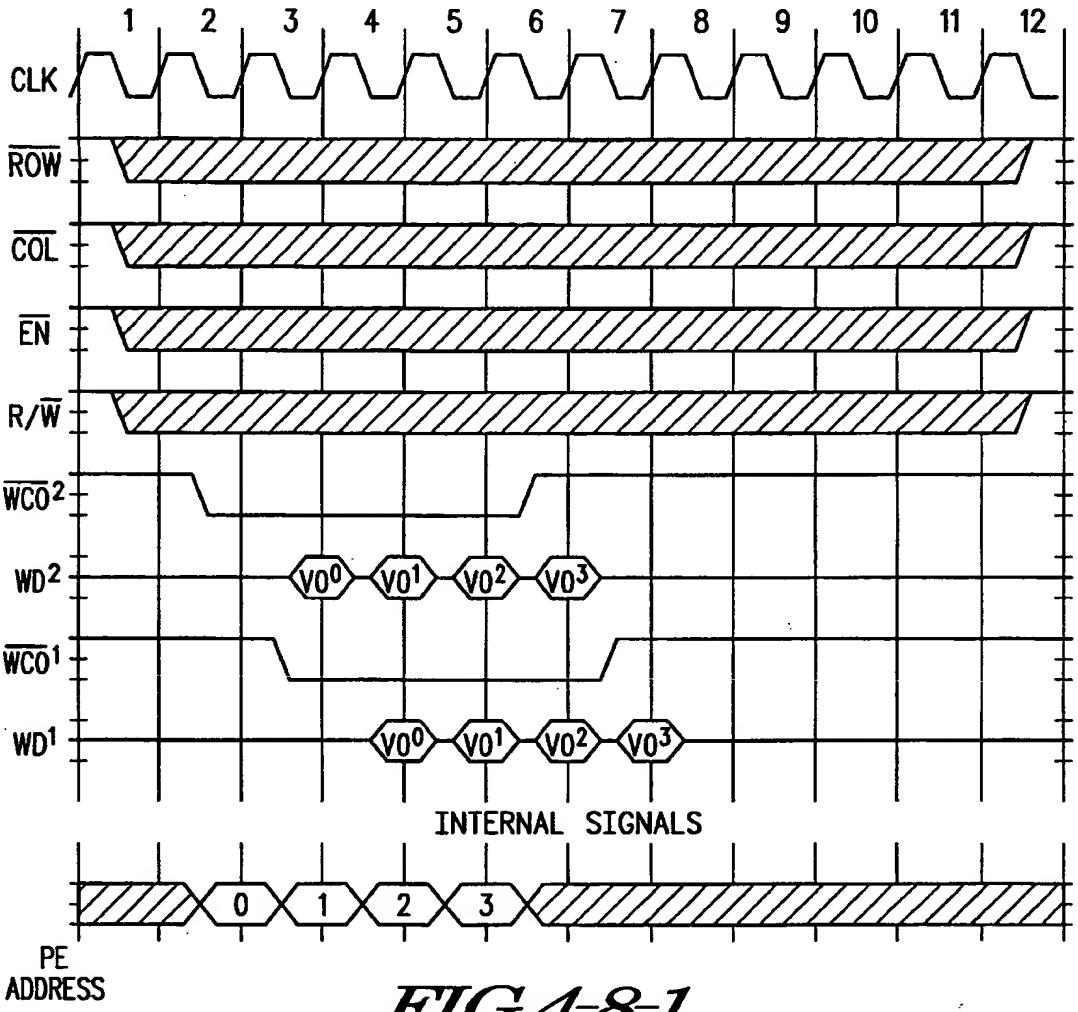


FIG.4-8-1

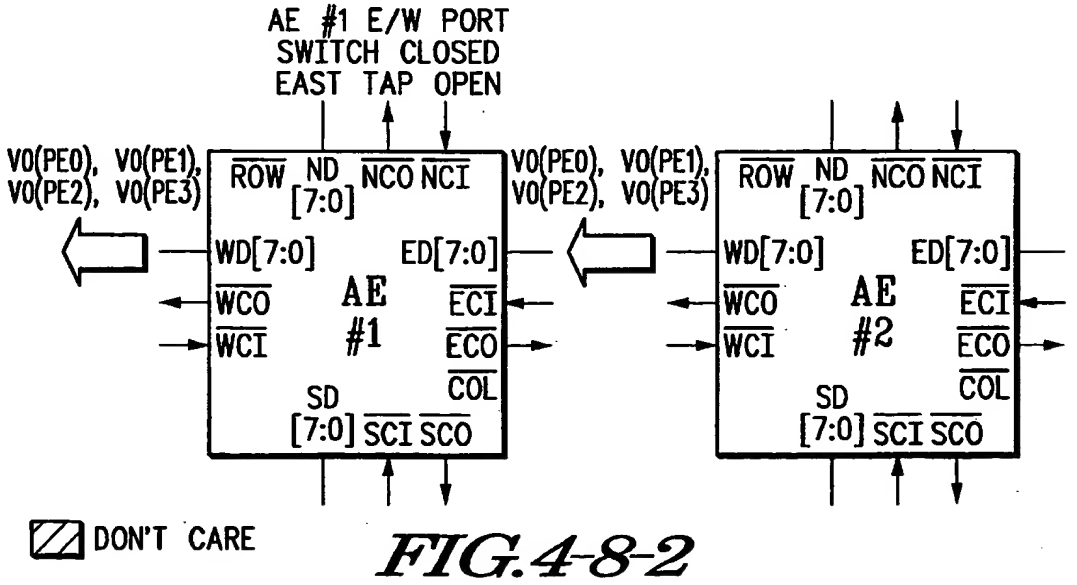


FIG.4-8-2

	Document ID	U	Title	Current OR
1	US 20020112146 A1	☐	Method and apparatus for synchronizing load operation	712/219
2	US 6047122 A	☐	System for method for performing a context switch operation in a massively parallel computer system	709/108
3	US 20030084259 A	☒	Memory fence and load fence micro-architectural implementation method for speech synthesis, involves dispatching load fencing instruction to cache controller, after removing old loads from memory subsystem	
4	KR 2001047533 A	☒	Synchronous memory device	

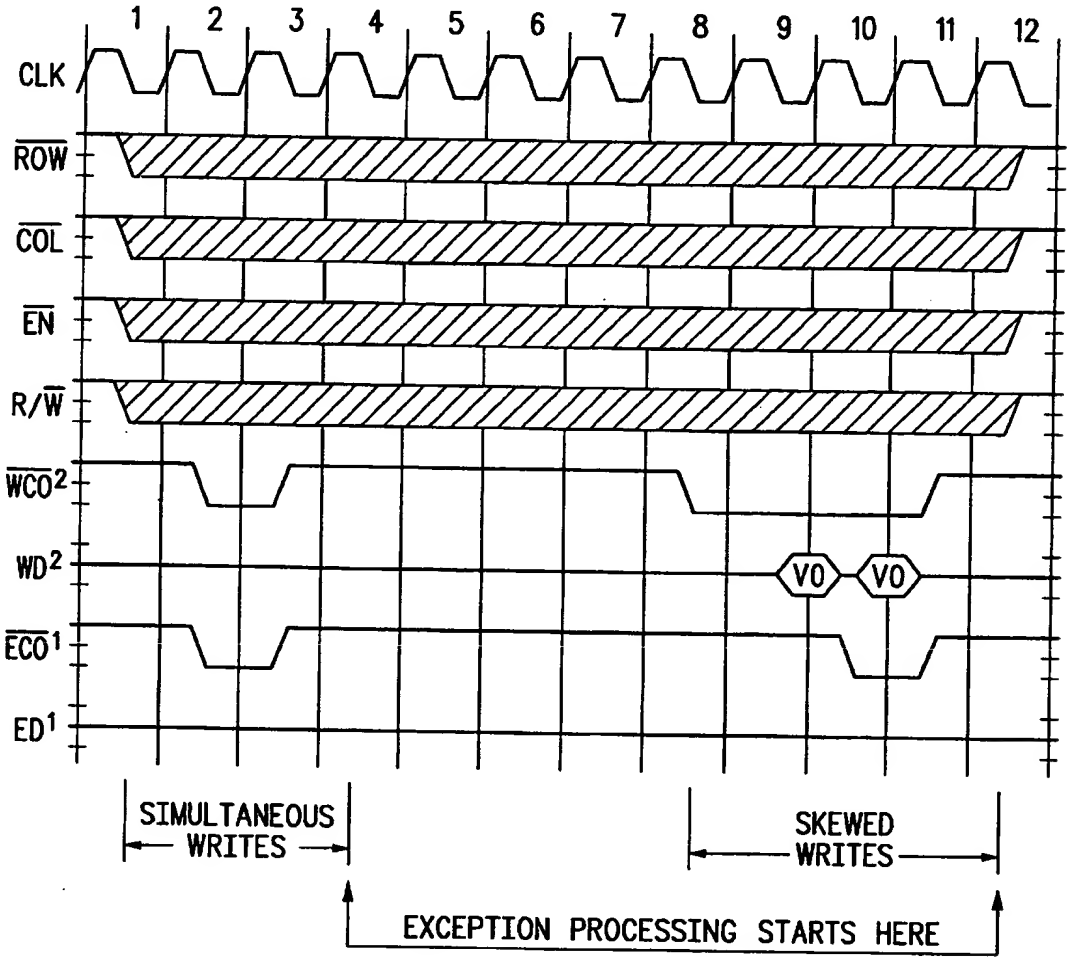


FIG.4-10-1

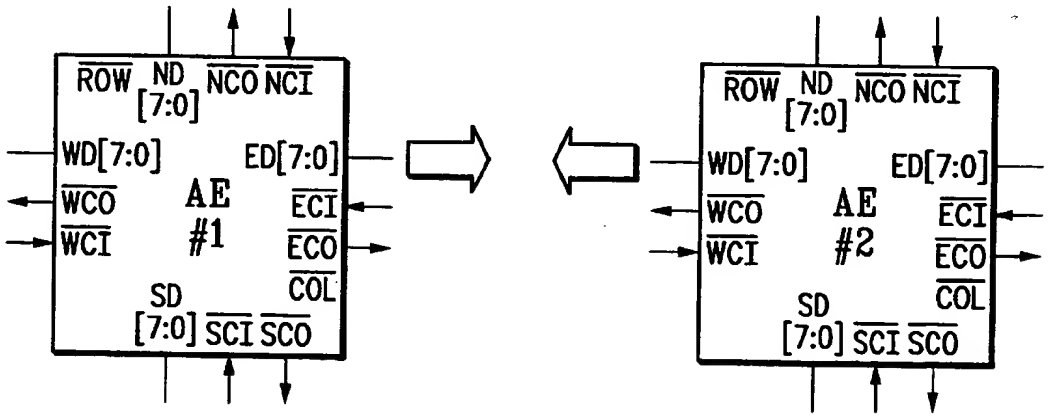


FIG.4-10-2

	Document ID	U	Title	Current OR
1	US 20030034544 A1	☐	Microcomputer	257/523
2	US 20010037434 A1	☒	Store to load forwarding using a dependency link file	711/146
3	US 6651151 B2	☒	MFENCE and LFENCE micro-architectural implementation method and system	711/163
4	US 6629271 B1	☒	Technique for synchronizing faults in a processor having a replay system	714/49
5	US 6609192 B1	☒	System and method for asynchronously overlapping storage barrier operations with old and new storage operations	712/216
6	US 6549990 B2	☒	Store to load forwarding using a dependency link file	711/146
7	US 6546462 B1	☒	CLFLUSH micro-architectural implementation method and system	711/135
8	US 6473837 B1	☒	Snoop resynchronization mechanism to preserve read ordering	711/146
9	US 6473832 B1	☒	Load/store unit having pre-cache and post-cache queues for low latency load memory operations	711/118
10	US 6466988 B1	☒	Multiprocessor synchronization and coherency control system	709/248
11	US 6427193 B1	☒	Deadlock avoidance using exponential backoff	711/146
12	US 6415360 B1	☒	Minimizing self-modifying code checks for uncacheable memory types	711/139
13	US 6414368 B1	☒	Microcomputer with high density RAM on single chip	257/523
14	US 6311261 B1	☒	Apparatus and method for improving superscalar processors	712/23
15	US 6266744 B1	☒	Store to load forwarding using a dependency link file	711/146
16	US 6189088 B1	☒	Forwarding stored dara fetched for out-of-order load/read operation to over-taken operation read-accessing same memory location	712/216
17	US 6167509 A	☒	Branch performance in high speed processor	712/237
18	US 6119204 A	☒	Data processing system and method for maintaining translation lookaside buffer TLB coherency without enforcing complete instruction serialization	711/141
19	US 6112019 A	☒	Distributed instruction queue	712/214
20	US 6076158 A	☒	Branch prediction in high-performance processor	712/230
21	US 6073210 A	☒	Synchronization of weakly ordered write combining operations using a fencing mechanism	711/118
22	US 5995746 A	☒	Byte-compare operation for high-performance processor	712/220

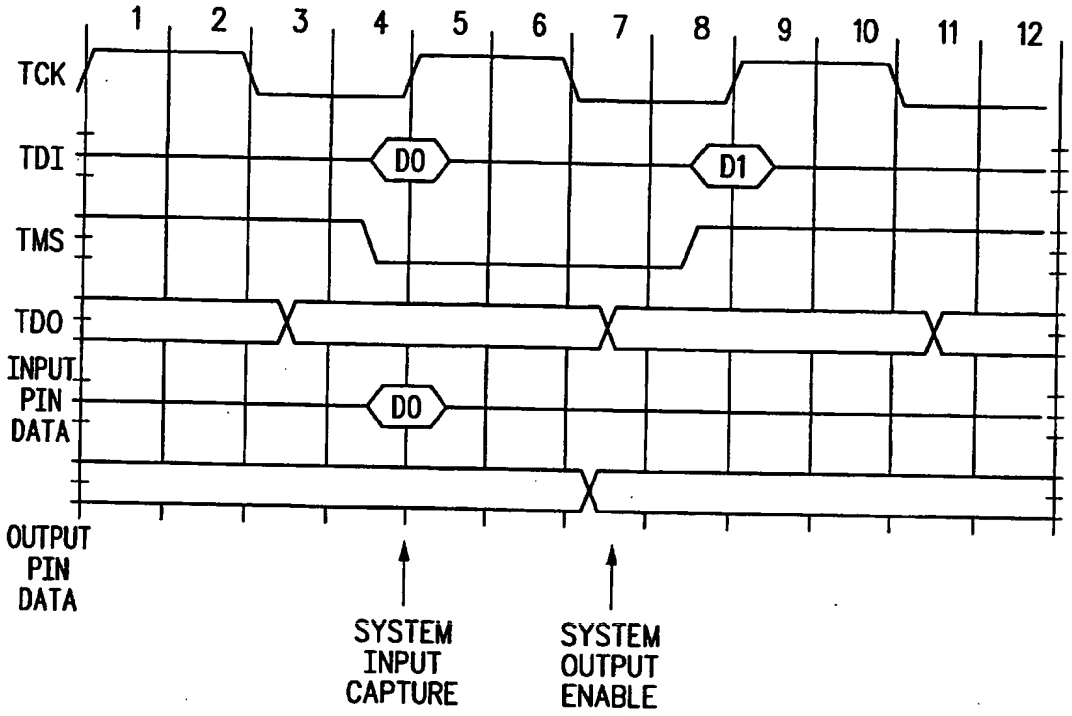


FIG.4-16

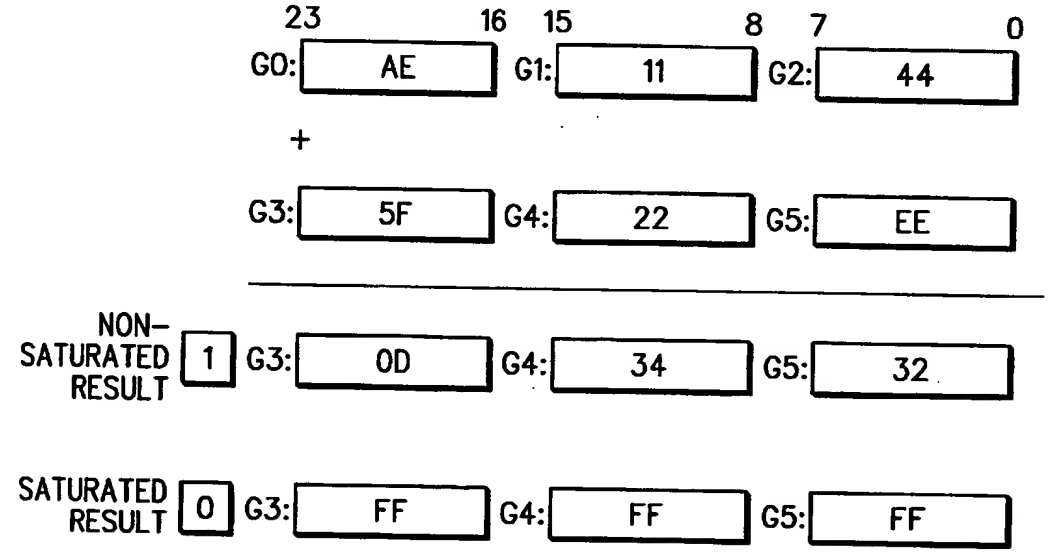


FIG.5-1-1

	Document ID	U	Title	Current OR
23	US 5881262 A	☒	Method and apparatus for blocking execution of and storing load operations during their execution	712/216
24	US 5826109 A	☒	Method and apparatus for performing multiple load operations to the same memory location in a computer system	710/39
25	US 5778423 A	☒	Prefetch instruction for improving performance in reduced instruction set processor	711/118
26	US 5778245 A	☒	Method and apparatus for dynamic allocation of multiple buffers in a processor	712/23
27	US 5724536 A	☒	Method and apparatus for blocking execution of and storing load operations during their execution	712/216
28	US 5694574 A	☒	Method and apparatus for performing load operations in a computer system	711/140
29	US 5568624 A	☒	Byte-compare operation for high-performance processor	712/223
30	US 5506437 A	☒	Microcomputer with high density RAM in separate isolation well on single chip	257/373
31	US 5491359 A	☒	Microcomputer with high density ram in separate isolation well on single chip	257/373
32	US 5469551 A	☒	Method and apparatus for eliminating branches using conditional move instructions	712/239
33	US 5454091 A	☒	Virtual to physical address translation scheme with granularity hint for identifying subsequent pages to be accessed	711/203
34	US 5452467 A	☒	Microcomputer with high density ram in separate isolation well on single chip	716/1
35	US 5424969 A	☒	Product-sum operation unit	708/603
36	US 5410682 A	☒	In-register data manipulation for unaligned byte write using data shift in reduced instruction set processor	712/300
37	US 5377336 A	☒	Improved method to prefetch load instruction data	712/207
38	US 5367705 A	☒	In-register data manipulation using data shift in reduced instruction set processor	712/41
39	US 5243698 A	☒	Microcomputer	709/201
40	US 5193167 A	☒	Ensuring data integrity by locked-load and conditional-store operations in a multiprocessor system	711/163
41	US 5150470 A	☒	Data processing system with instruction queue having tags indicating outstanding data status	712/217
42	US 5031092 A	☒	Microcomputer with high density ram in separate isolation well on single chip	711/163
43	US 4975960 A	☒	Electronic facial tracking and detection system and method and apparatus for automated speech recognition	704/251
44	US 4967326 A	☒	Microcomputer building block	712/21
45	US 4819151 A	☒	Microcomputer	709/106

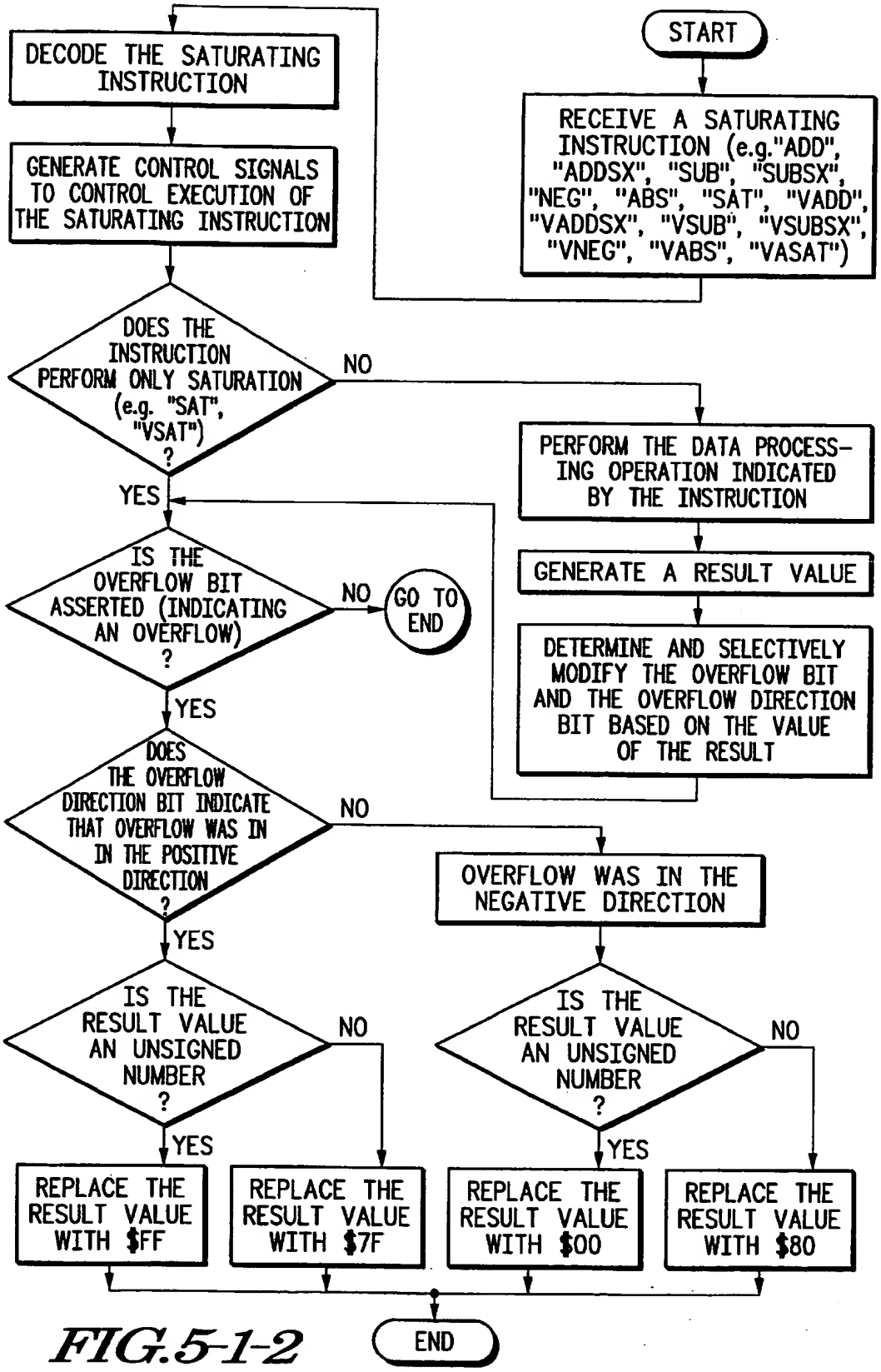

START
RECEIVE A SATURATING INSTRUCTION (e.g."ADD", "ADDSX", "SUB", "SUBSX", "NEG", "ABS", "SAT", "VADD", "VADDSX", "VSUB", "VSUBSX", "VNEG", "VABS", "VASAT")
DECODE THE SATURATING INSTRUCTION
GENERATE CONTROL SIGNALS TO CONTROL EXECUTION OF THE SATURATING INSTRUCTION
DOES THE INSTRUCTION PERFORM ONLY SATURATION (e.g. "SAT", "VSAT") ?
NO
YES
PERFORM THE DATA PROCESSING OPERATION INDICATED BY THE INSTRUCTION
GENERATE A RESULT VALUE
DETERMINE AND SELECTIVELY MODIFY THE OVERFLOW BIT AND THE OVERFLOW DIRECTION BIT BASED ON THE VALUE OF THE RESULT
IS THE OVERFLOW BIT ASSERTED (INDICATING AN OVERFLOW) ?
NO
GO TO END
YES
DOES THE OVERFLOW DIRECTION BIT INDICATE THAT OVERFLOW WAS IN IN THE POSITIVE DIRECTION ?
NO
YES
OVERFLOW WAS IN THE NEGATIVE DIRECTION
IS THE RESULT VALUE AN UNSIGNED NUMBER ?
NO
YES
IS THE RESULT VALUE AN UNSIGNED NUMBER ?
NO
YES
REPLACE THE RESULT VALUE WITH $FF
REPLACE THE RESULT VALUE WITH $7F
REPLACE THE RESULT VALUE WITH $00
REPLACE THE RESULT VALUE WITH $80
END

FIG.5-1-2

	Document ID	U	Title	Current OR
46	US 4724517 A	☒	Microcomputer with prefixing functions	712/210
47	US 4704678 A	☒	Function set for a microcomputer	709/106
48	US 4680698 A	☒	High density ROM in separate isolation well on single with chip	712/37
49	US 4214269 A	☒	Real time digital scan converter	348/442
50	US 3984814 A	☒	Retry method and apparatus for use in a magnetic recording and reproducing system	714/16
51	US 3790958 A	☒	DATA COMMUNICATION TERMINAL	710/65
52	US 3728710 A	☐	CHARACTER DISPLAY TERMINAL	345/17

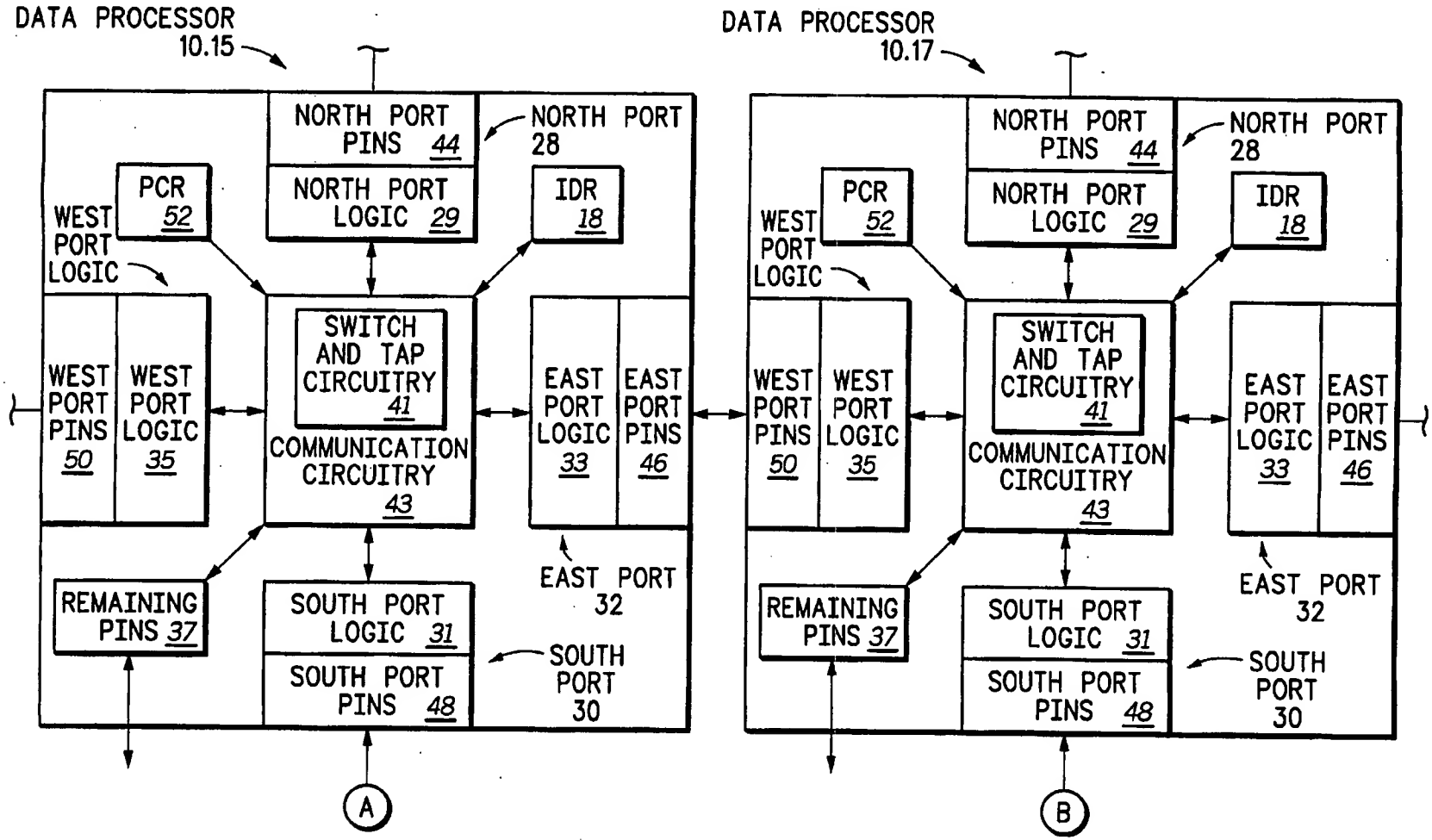
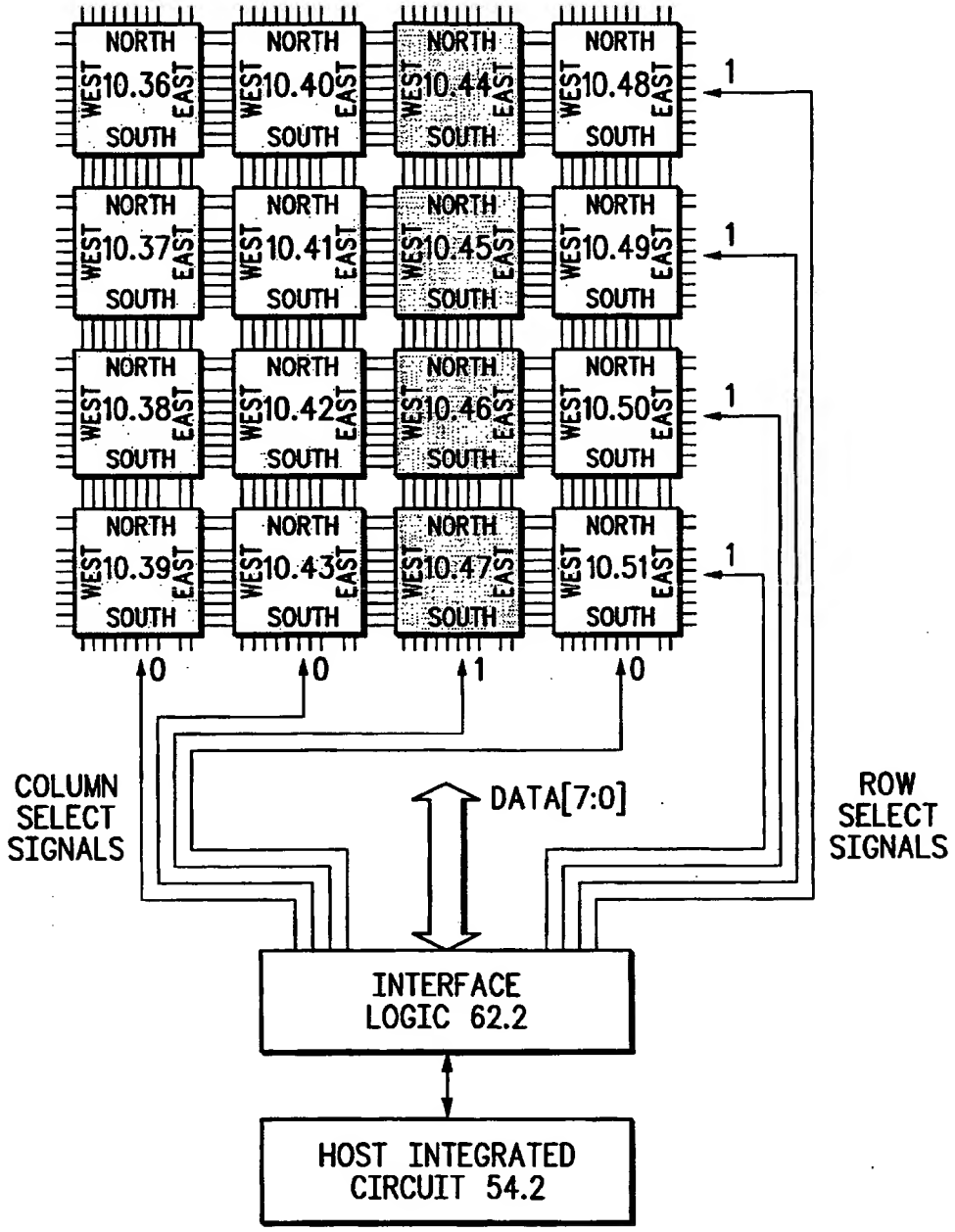

DATA PROCESSOR
10.15
NORTH PORT PINS 44
NORTH PORT LOGIC 29
NORTH PORT 28
PCR 52
IDR 18
WEST PORT LOGIC
WEST PORT PINS 50
WEST PORT LOGIC 35
SWITCH AND TAP CIRCUITRY 41
COMMUNICATION CIRCUITRY 43
EAST PORT LOGIC 33
EAST PORT PINS 46
EAST PORT 32
REMAINING PINS 37
SOUTH PORT LOGIC 31
SOUTH PORT PINS 48
SOUTH PORT 30
A
DATA PROCESSOR
10.17
NORTH PORT PINS 44
NORTH PORT LOGIC 29
NORTH PORT 28
PCR 52
IDR 18
WEST PORT LOGIC
WEST PORT PINS 50
WEST PORT LOGIC 35
SWITCH AND TAP CIRCUITRY 41
COMMUNICATION CIRCUITRY 43
EAST PORT LOGIC 33
EAST PORT PINS 46
EAST PORT 32
REMAINING PINS 37
SOUTH PORT LOGIC 31
SOUTH PORT PINS 48
SOUTH PORT 30
B

FIG.5-2-1

	Document ID	U	Title	Current OR
1	US 20030084259 A1	☐	MFENCE and LFENCE micro-architectural implementation method and system	711/163
2	US 20020199067 A1	☒	System and method for high performance execution of locked memory instructions in a system with distributed memory and a restrictive memory model	711/145
3	US 20020112146 A1	☐	Method and apparatus for synchronizing load operation	712/219
4	US 20020087810 A1	☒	System and method for high performance execution of locked memory instructions in a system with distributed memory and a restrictive memory model	711/145
5	US 20010034819 A1	☒	Interleaved data path and output management architecture for an interleaved memory and load pulser circuit for outputting the read data	711/157
6	US 20010033245 A1	☒	Interleaved memory device for burst type access in synchronous read mode with the two semi-arrays independently readable in random access asynchronous mode	341/200
7	US 6651151 B2	☒	MFENCE and LFENCE micro-architectural implementation method and system	711/163
8	US 6611900 B2	☒	System and method for high performance execution of locked memory instructions in a system with distributed memory and a restrictive memory model	711/145
9	US 6587913 B2	☒	Interleaved memory device for burst type access in synchronous read mode with the two semi-arrays independently readable in random access asynchronous mode	711/5
10	US 6480818 B1	☒	Debugging techniques in a multithreaded environment	703/26
11	US 6473402 B1	☒	Communications link interconnecting service control points of a load sharing group for traffic management control	370/236
12	US 6470431 B2	☒	Interleaved data path and output management architecture for an interleaved memory and load pulser circuit for outputting the read data	711/157
13	US 6463511 B2	☒	System and method for high performance execution of locked memory instructions in a system with distributed memory and a restrictive memory model	711/145
14	US 6370625 B1	☒	Method and apparatus for lock synchronization in a microprocessor system	711/152
15	US 6353829 B1	☒	Method and system for memory allocation in a multiprocessing environment	707/100
16	US 6347349 B1	☒	System for determining whether a subsequent transaction may be allowed or must be allowed or must not be allowed to bypass a preceding transaction	710/62
17	US 6314471 B1	☒	Techniques for an interrupt free operating system	710/5
18	US 6260088 B1	☒	Single integrated circuit embodying a risc processor and a digital signal processor	710/100
19	US 6073210 A	☒	Synchronization of weakly ordered write combining operations using a fencing mechanism	711/118
20	US 6070003 A	☒	System and method of memory access in apparatus having plural processors and plural memories	710/317
21	US 6047122 A	☐	System for method for performing a context switch operation in a massively parallel computer system	709/108

STOP MODE SOUTH STREAM ACCESSES



DATA PROCESSING SYSTEM 55.2

FIG.5-6

	Document ID	U	Title	Current OR
22	US 6038584 A	☒	Synchronized MIMD multi-processing system and method of operation	709/248
23	US 6016270 A	☒	Flash memory architecture that utilizes a time-shared address bus scheme and separate memory cell access paths for simultaneous read/write operations	365/185.11
24	US 5933624 A	☒	Synchronized MIMD multi-processing system and method inhibiting instruction fetch at other processors while one processor services an interrupt	709/400
25	US 5905967 A	☒	Timing generator with multiple coherent synchronized clocks	702/118
26	US 5881272 A	☒	Synchronized MIMD multi-processing system and method inhibiting instruction fetch at other processors on write to program counter of one processor	709/400
27	US 5809288 A	☒	Synchronized MIMD multi-processing system and method inhibiting instruction fetch on memory access stall	709/400
28	US 5768609 A	☒	Reduced area of crossbar and method of operation	712/11
29	US 5758195 A	☒	Register to memory data transfers with field extraction and zero/sign extension based upon size and mode data corresponding to employed address register	712/300
30	US 5696913 A	☒	Unique processor identifier in a multi-processing system having plural memories with a unified address space corresponding to each processor	710/317
31	US 5613146 A	☒	Reconfigurable SIMD/MIMD processor using switch matrix to allow access to a parameter memory by any of the plurality of processors	712/20
32	US 5606520 A	☒	Address generator with controllable modulo power of two addressing capability	708/491
33	US 5592405 A	☒	Multiple operations employing divided arithmetic logic unit and multiple flags register	708/518
34	US 5560029 A	☒	Data processing system with synchronization coprocessor for multiple threads	712/25
35	US 5522083 A	☒	Reconfigurable multi-processor operating in SIMD mode with one processor fetching instructions for use by remaining processors	712/22
36	US 5510689 A	☒	Air gap flux measurement using stator third harmonic voltage	318/809
37	US 5471592 A	☒	Multi-processor with crossbar link of processors and memories and method of operation	709/213
38	US 5430850 A	☒	Data processing system with synchronization coprocessor for multiple threads	709/314
39	US 5410649 A	☒	Imaging computer system and network	345/505
40	US 5371896 A	☒	Multi-processor having control over synchronization of processors in mind mode and method of operation	712/20
41	US 5339447 A	☒	Ones counting circuit, utilizing a matrix of interconnected half-adders, for counting the number of ones in a binary string of image data	377/82
42	US 5272429 A	☒	Air gap flux measurement using stator third harmonic voltage and uses	318/808
43	US 5239654 A	☒	Dual mode SIMD/MIMD processor providing reuse of MIMD instruction memories as data memories when operating in SIMD mode	712/20
44	US 5226125 A	☒	Switch matrix having integrated crosspoint logic and method of operation	710/317

RUN MODE DATA OPERATIONS

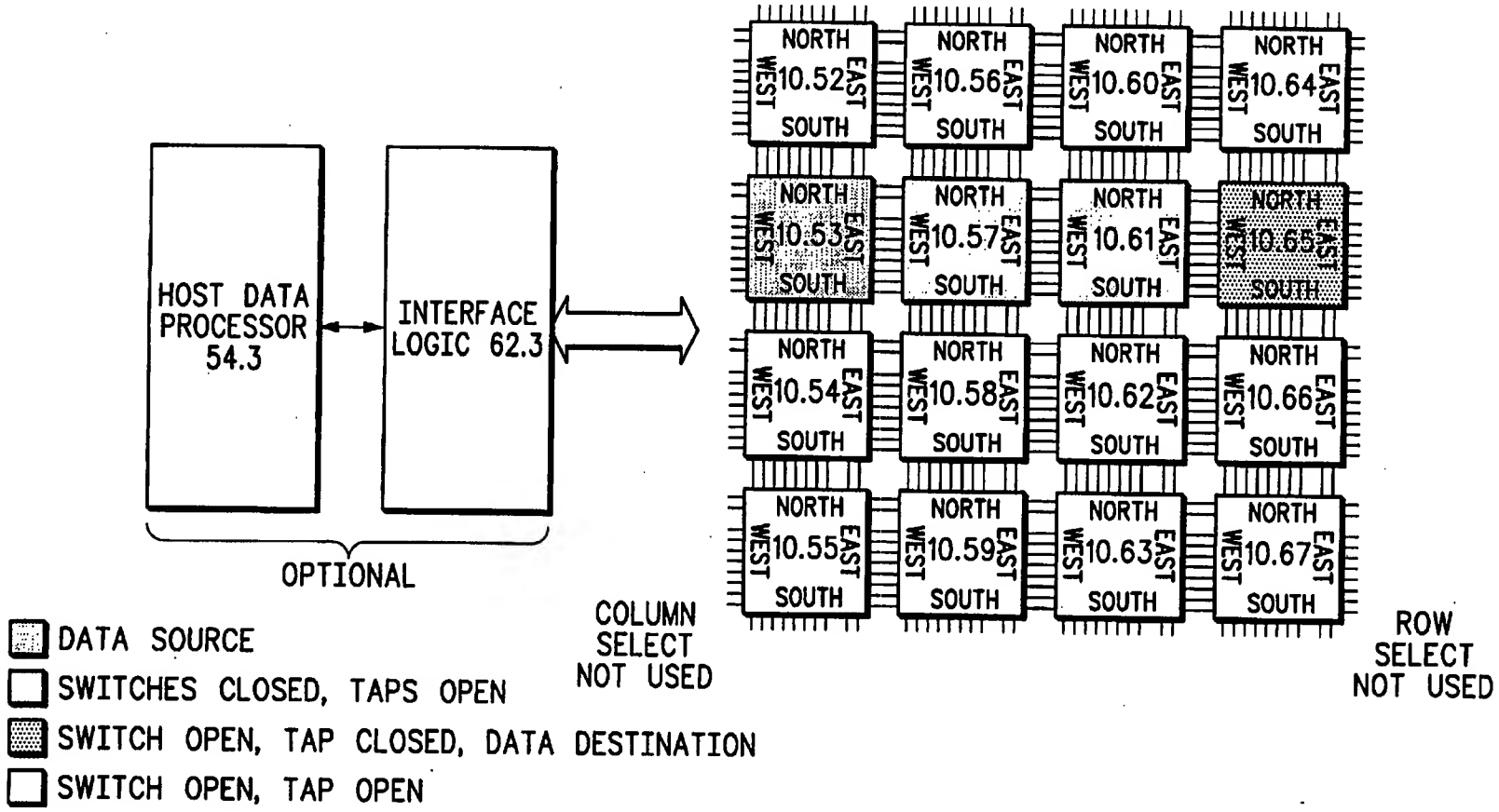


FIG.5-7

	Document ID	U	Title	Current OR
45	US 5212777 A	☒	Multi-processor reconfigurable in single instruction multiple data (SIMD) and multiple instruction multiple data (MIMD) modes and method of operation	712/229
46	US 5197140 A	☒	Sliced addressing multi-processor and method of operation	711/220

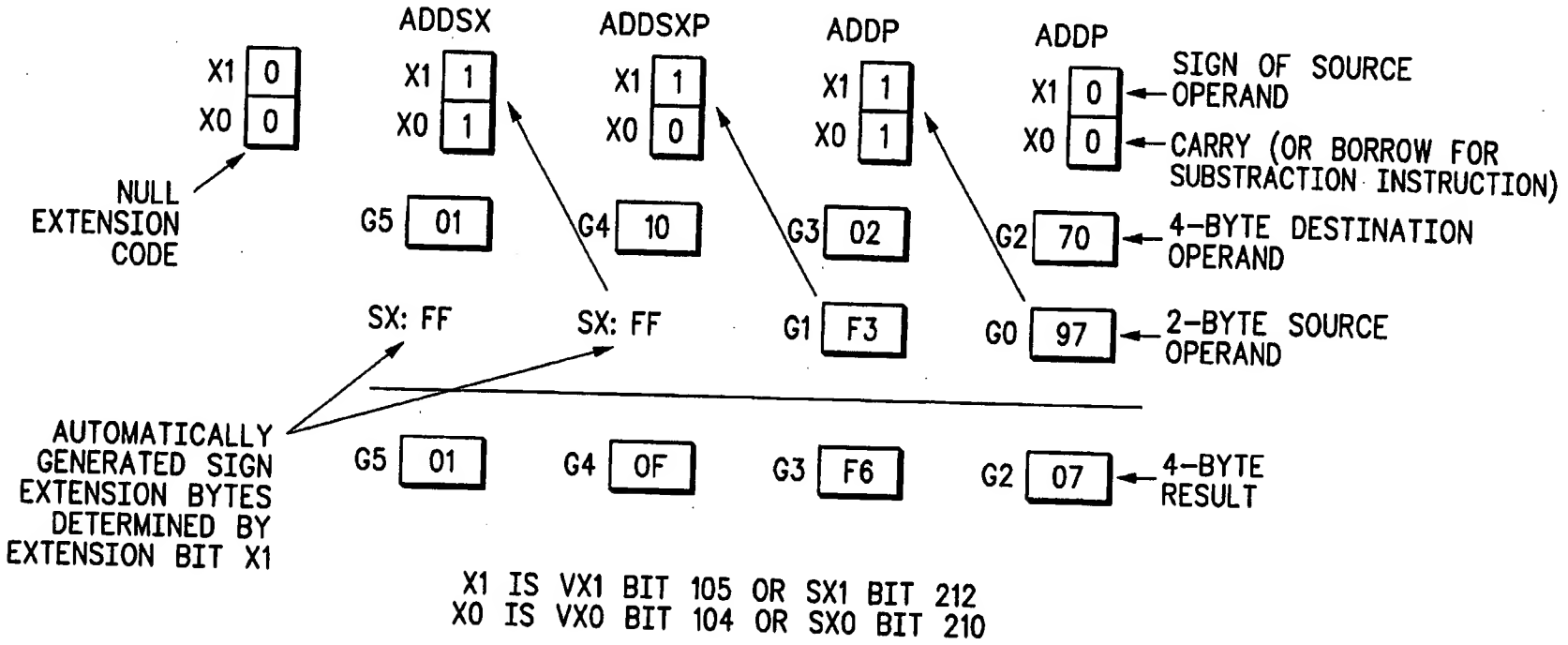

ADDSX
ADDSXP
ADDP
ADDP
X1 0
X0 0
NULL EXTENSION CODE
X1 1
X0 1
X1 1
X0 0
X1 1
X0 1
X1 0
X0 0
SIGN OF SOURCE OPERAND
CARRY (OR BORROW FOR SUBSTRACTION INSTRUCTION)
G5 01
G4 10
G3 02
G2 70
4-BYTE DESTINATION OPERAND
SX: FF
SX: FF
G1 F3
G0 97
2-BYTE SOURCE OPERAND
AUTOMATICALLY GENERATED SIGN EXTENSION BYTES DETERMINED BY EXTENSION BIT X1
G5 01
G4 0F
G3 F6
G2 07
4-BYTE RESULT
X1 IS VX1 BIT 105 OR SX1 BIT 212
X0 IS VX0 BIT 104 OR SX0 BIT 210

FIG.5-8